IES de Teis
Avda. de Galicia, 101
36216 – Vigo

Tfno.: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

Unidad didáctica 1. Big Data.

Tabla de contenido

1.- Big Data

Una primera definición del concepto Big Data sería "la gestión relacionada con el almacenamiento y uso de una cantidad tan grande de datos que no puede ser realizada por sistemas y métodos tradicionales".

Los problemas que surgen al intentar manejar una gran cantidad de datos no son nuevos y siempre han requerido de nuevas tecnologías. Dentro de la informática podríamos citar la máquina tabuladora de Hollerith como uno de los primeros ejemplos.

En Estados Unidos se realiza el censo cada 10 años. En 1880 se usaron 7 años para elaborar los resultados y se estimaba que para el siguiente se necesitarían más de 10 años. Herman Hollerith diseñó una máquina tabuladora y adaptó las preguntas del censo para que todas fueran de tipo booleano y consiguió que los 60 millones de registros del censo de 1890 estuvieran listos en 2 años.

No hay un momento claro del nacimiento del término **Big Data,** pero podríamos situarlo en la década de 1990 con el nacimiento de la www y los primeros buscadores de páginas web. La cantidad de información que había que manejar excedía la capacidad de una sola máquina.

A pesar de que la tecnología hardware permitía cada vez mayores y más rápidos dispositivos de almacenamiento, la cantidad de datos que se generaban superaban con creces cualquier intento de ampliar y mejorar los equipos. A solucionar la falta de procesamiento o almacenaje mediante la mejora hardware de un equipo servidor se le llama **escalado vertical**.

Un escalado vertical es una opción poco económica puesto que los componentes de última generación siempre tienen los precios más caros, además estas ampliaciones solo se pueden hacer hasta cierto punto porque cualquier arquitectura tiene unos límites físicos propios.

Para afrontar el problema del almacenamiento se optó por una tecnología que ya existía, los sistemas de ficheros distribuidos como por ejemplo NFS. La respuesta a la necesidad de almacenamiento masivo fue que un conjunto de servidores funcionase como una sola entidad lógica y, de esta manera, ampliar capacidad consistía en añadir un nuevo nodo a la red. A este enfoque se le llama **escalado horizontal**.

A principios del siglo XXI, llega la web 2.0, la burbuja de las "punto com", las redes sociales, mensajería instantánea, IoT, sensores y muchas otras fuentes de datos donde lo importante no es solo el post, el like o el mensaje que se transmite, sino que también desde donde, cuando, quien, a quien, qué sistema operativo tiene el usuario, cuanto tiempo pasa en la web, en definitiva, los **metadatos**.

La cantidad de información que se genera entra en una escala difícil de visualizar. Piensa en la cantidad de información que generaba una persona en 1990. Probablemente se limitaba a sus movimientos

bancarios, alguna llamada telefónica y puede que algún asunto médico. Hoy en día, la mayoría de las personas generan una cantidad de datos muy grande en forma de mensajería instantánea, ubicación, monitorización de salud, contador de pasos, llamadas, streaming, navegación por internet y muchas más.

De la misma manera que la información que proporciona el censo se usa para distribuir recursos y tomar decisiones en un país, con los metadatos también se pueden establecer perfiles de todo tipo.

Un motón de datos almacenados por si solos no son muy útiles, necesitan procesarse para obtener información relevante. El nuevo problema que acarrea una cantidad tan grande de datos almacenados es su proceso ya que no se tarda lo mismo en recorrer 1 megabyte de datos que 1 petabyte de datos. Resultaba necesario distribuir el procesamiento en paralelo y unir las partes para ofrecer un resultado final.

Con el problema del almacenamiento y el procesamiento sobre la mesa, las empresas con más interés en el asunto sentaron las bases de lo que hoy llamamos Big Data.

Usando el ecosistema de Hadoop se consigue almacenamiento distribuido con HDFS y también procesamiento paralelo con YARN y distintos motores como MapReduce.

1.1.- Las 5 Vs

Las múltiples definiciones que podemos encontrar de Big Data tienen en común que giran en torno al concepto de dato y de ahí ha surgido un juego de palabras que empiezan por la letra "v". En su origen fueron 3 V´s que debían cumplir todos los problemas que fueran considerados Big Data.

- Volume
- Velocity
- Variety

Hay un consenso amplio en añadir otras 2 V´s que tienen sentido en Big Data y que hacen referencia a la forma de procesar y resultado final, estas son:

- Veracity
- Value

Mas allá de estas V´s entramos en una zona de fantasía donde parece que solo se trata de dar un significado relacionado a todas las palabras que empiezan por V. Alguna de las V´s que encontramos aquí es "Viral", "Virtuoso" o "Viscoso". Puedes verlas una buena parte de ellas en la web: https://www.kdnuggets.com/2017/04/42-vs-big-data-data-science.html

1.1.1.- Volumen

Hace referencia a la cantidad de datos que se manejan en el problema. Normalmente consideramos que el volumen de datos con los que se trabaja en Big Data no tiene cabida en un único equipo por muy potente que este sea.

En esta parte debemos conocer las unidades de medida de capacidad.

- **Bit**. Es la unidad mínima de capacidad en informática. Solo puede almacenar dos valores, o cero o uno. De ahí su nombre (binary + digit = bit). Con los bits podemos operar matemáticamente al igual que con nuestros números enteros. Además, si codificamos esos dos valores que puede guardar como "Verdadero" o "Falso" podremos aplicar álgebra de Boole y reglas de lógica proposicional
- **Byte**. Es la agrupación de 8 bits. Son 8 porque eran las cifras necesarias para codificar los 255 caracteres de la tabla ASCII. Cuando se habla de tamaño de un archivo o disco siempre se usa esta unidad. Hablamos en esta unidad cuando nos referimos a algo realmente pequeño como un dato o incluso un registro. Por ejemplo, un dato decimal ocupa unos 4 bytes.
- **Kilobyte**. Es la agrupación de 1000 bytes. En términos informáticos es no es correcto puesto que trabajamos con el sistema binario. En realidad, es algo a lo que estamos acostumbrados porque trabajamos con sistemas numéricos distintos a lo largo del día en los que la agrupación decimal (10, 100, 1000, ...) no tiene sentido. Agrupamos los segundos en bloques de 60 para hacer un minuto, agrupamos los minutos en bloques de 60 para hacer una hora, agrupamos las horas en bloques de 24 para hacer un día, agrupamos los días en bloques de 7 para hacer una semana... Usamos esta unidad para referirnos a archivos pequeños de texto plano como por ejemplo los de configuración que podemos encontrar en */etc/*
- **Megabyte**. Es la agrupación de 1000 Kilobytes. En la actualidad esta es la unidad más usada en informática a nivel de usuario. Una canción, una foto, un archivo pueden pesar entre 2 y 15 megabytes.
- **Gigabyte**. Es la agrupación de 1000 Megabytes. Usaremos esta unidad para referirnos a archivos muy grandes. Por ejemplo, una película o una imagen de disco pueden ocupar entre 2 y 25 gigabytes.
- **Terabyte**. Es la agrupación de 1000 Gigabytes. Un usuario normal no maneja archivos tan pesados así que usará esta unidad para hacer referencia a la capacidad de sus discos. Un disco de buena capacidad tendrá entre 1 y 4 terabytes.
- **Petabyte**. Es la agrupación de 1000 Terabytes. Esta escala es la que marca la frontera entre informática de escritorio y la parte de big data.

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno.: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

Binario		
Simbolo	Prefijo	Factor
Ki	Kibi	2^{10}
Mi	Mebi	2^{20}
Gi	Gibi	2^{30}
Ti	Tebi	2^{40}
Pi	Pebi	2^{50}
Ei	Exbi	2^{60}
Zi	Zebi	2^{70}
Yi	Yobi	2^{80}

Decimal y diferencia con binario			
Factor	Prefijo	Bin÷Dec	Error
10^3	Kilo	1.024	2.4%
10^6	Mega	1.049	4.9%
10^9	Giga	1.074	7.4%
10^{12}	Tera	1.100	10.0%
10^{15}	Peta	1.126	12.6%
10^{18}	Exa	1.153	15.3%
10^{21}	Zetta	1.181	18.1%
10^{24}	Yotta	1.209	20.9%

En informática usamos el sistema binario por lo que las agrupaciones en base 10 no tiene mucho sentido. La nomenclatura del sistema internacional nos obliga a que el prefijo "kilo" siempre signifique 1.000 así que nos hemos inventado el prefijo "kibi" (kilo binario) para indicar agrupaciones de 1024.

- **Kibibyte**. Es la agrupación en 1024 (2^{10}) bytes. Esta es la medida informática exacta. Normalmente abusamos del lenguaje y usamos el término kilobyte para referirnos a la más apropiada en cada momento.
- **Mebibyte**. Es la agrupación de 1024 kibibytes.
- **Gibibyte**. Es la agrupación de 1024 mebibytes.
- **Tebibyte**. Es la agrupación de 1024 gibibytes.

Ahora que tenemos claras las unidades de medida es el momento de enfocar a qué se refiere esta V de "volumen" y para ello estaría bien hacer una búsqueda en internet sobre lo que ocurre en internet durante un día.

Redes sociales como Facebook comparte más de 200.00 fotos entre sus 40 millones de usuarios, Instagram sube 60.000 de sus 60 millones de usuarios, Tiktok, Twitter, Discord, Zoom, Microsoft Teams y muchas otras también generan muchísimo tráfico.

También se genera información con la navegación de solamente descarga como las más de 5 millones de búsquedas en Google, las más de 600.000 horas de vídeos reproducidos de Youtube, streaming de plataformas como PrimeVideo, Netflix, Disney+, HBO...

Hoy en día las compras online también son una fuente de datos importante. Amazon, Aliexpress, marketplaces de distintas empresas generan muchísimo movimiento. Solo en bizum y paypal se genera más de 1 millón de dólares en operaciones de compra.

En internet también hay mucho tráfico que no se ve pero que existe como el de las llamadas telefónicas, los datos que generan todos los sensores del internet de las cosas o los movimientos bancarios.

Lo impresionante es que estos números se refieren a eventos que suceden de media en un solo minuto. Piensa también que solo hemos nombrado los hechos y no hemos tenido en cuenta los metadatos que se generan en cada una de esas operaciones.

1.1.2.- Velocidad

El volumen de datos con el que se trabaja en Big Data ya da pistas de la velocidad a la que se deben procesar los datos y a esto es a lo que se refiere esta "v" de "velocidad".

Big Data no tendría mucho sentido si no pudiera proporcionar resultados en tiempo real. En base a los datos almacenados, los bancos y distintas empresas pueden detectar cuando una operación parece fraudulenta. Esa detección debería realizarse en tiempo real. En cambio, una actualización del número de likes de un post o el número exacto de seguidores en una red social no es algo que requiera una velocidad de tiempo real.

Por ello en big data existen distintas estrategias de procesamiento en streaming, tiempo real, batch y otras que veremos a lo largo del curso.

1.1.3.- Variedad

Estamos acostumbrados a trabajar con datos bien definidos y rápidamente se nos viene a la cabeza ideas y conceptos de las bases de datos relacionales donde tenemos campos que guardan propiedades concretas, registros compuestos por campos que definen una instancia de una entidad y tablas que guardan los registros que se relacionan con ese tipo de entidad.

A este tipo de dato se le llama **dato estructurado** porque está definido mediante campos estrictos y claramente definidos como fechas, números y cadenas de texto. En general son los tipos de datos ideales para ser usados con lenguajes de consulta como SQL. Los clientes de una empresa o los movimientos bancarios son dos ejemplos típicos, aunque aquí también encontramos archivos con logs y otros.

Hoy en día se nos ofrece la posibilidad de no solamente trabajar con datos de texto sino también con archivos multimedia como imágenes, audios o vídeos. A estos tipos de datos se les llama **dato no estructurado**. Se caracterizan por no tener una estructura interna bien definida, fija y porque su procesado requiere de técnicas mucho más complejas como el reconocimiento óptico de caracteres o el procesamiento de lenguaje natural. Hasta ahora este tipo de datos se almacenaban, pero no admitían mucho proceso automatizado. Un buen ejemplo son las radiografías y otras imágenes médicas que quedaban guardadas en el historial digital pero que era necesario que un profesional las interpretase.

En esta clasificación hay una zona gris en la que encontramos los **datos semiestructurados** que son aquellos que tienen parte estructurada y parte no estructurada. Un ejemplo lo encontramos en un correo electrónico donde tenemos una parte estructurada bien definida con campos como "Para", "Asunto", "Cuerpo" pero también tenemos una parte no estructurada como podrían ser los archivos adjuntos.

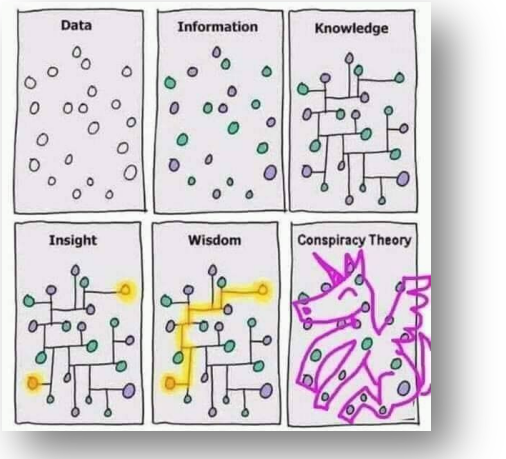
1.1.4.- Veracidad

Otra de las características que tienen los datos en Big Data es que deben ser veraces, correctos, ciertos y esto requiere de algún tipo de filtrado o procesamiento. Piensa que al guardar datos de un cliente es raro que tengamos "ruido" es decir, nunca encontraremos un "producto" en la tabla de "clientes" ni tampoco deberíamos encontrar clientes ininteligibles.

En cambio, los datos que generan los sensores sí pueden incluir valores falsos. Un sensor de temperatura puede entregar un -99 como valor puntual y este debe ser detectado.

1.1.5.- Valor

La última "v" que vamos a tratar es la relacionado con el valor que nos proporciona ese dato. No siempre los datos tienen el mismo valor, incluso el mismo dato no siempre tiene el mismo valor. Si hablamos de operaciones en bolsa son mucho más importantes los últimos valores que los de hace un año.



Saber seleccionar los datos con los que trataremos en Big Data también proporciona el valor que se busca.

Los **datos** en sí no aportan nada, una temperatura, un producto o cualquier registro no tienen entidad suficiente para obtener ninguna conclusión.

La **información** se obtiene a partir de un conjunto de datos. En una secuencia de temperatura sí podemos observar patrones y eventos extraordinarios.

El **conocimiento** se obtiene cuando tenemos la información suficiente entender los patrones y ser capaces de generar predicciones.

La **intuición** parte del conocimiento y permite seleccionar y correlacionar datos que están aparentemente desconectados.

El último paso es la **sabiduría** donde entendemos la cadena de sucesos que conlleva un dato.

Si pensamos en un ejemplo médico se entiende muy bien. Un dato vendría a ser tos o cualquier otro síntoma. Información sería el conjunto de los síntomas que refieren los pacientes durante un tiempo. El conocimiento nos permitiría predecir picos de enfermedades. La intuición permite que los médicos

estudien la correlación que hay entre fumar y padecer cáncer de pulmón. La sabiduría nos permite explicar los procesos que llevan a un fumador a finalmente padecer cáncer y como será su evolución.

Esta "v" suele nombrarse en un apartado llamado "De los eventos al valor" haciendo referencia a lo aquí visto.

1.2.- Qué conseguimos gracias a Big Data

Si tuviéramos que resumir el objetivo de Big Data en muy pocas palabras diríamos que consiste en almacenar y procesar un conjunto inmenso de datos para visualizarlos de manera que aporten respuestas que influyen en la toma de decisiones.

El ciclo del Big Data consiste en capturar datos, organizarlos e integrarlos para posteriormente analizarlos y actuar según los resultados. Encontramos un ejemplo de esto cuando Netflix nos ofrece alguna película o serie en función de nuestras selecciones o cuando al hacer una compra en Carrefour nos dan un vale para futuras compras. Ambas son acciones que se ejecutan en tiempo real pero que están basadas en muchos datos que antes llevaría días calcular.

En Big Data almacenamos los datos que de ninguna manera pueden almacenarse en una sola máquina. Además, se almacenan de manera que estén replicados por seguridad y también por eficiencia en el acceso y proceso. Gracias a esto se pueden analizar conjuntos de muchos datos que permiten:

- Optimizar operaciones de empresas
- Actuar inteligentemente basándonos en la evidencia de los datos.
- Identificar nuevos mercados,
- Realizar predicciones basándonos en modelos a partir de los datos.
- Detectar casos de fraude e impagos.
- Dar soporte a la toma de decisiones.
- Realizar descubrimientos científicos.
- Detección de enfermedades en función de los datos del historial y pruebas.
- Creación de nuevos fármacos.

2.- Clústers de computadoras

La magia de Big Data se logra fundamentalmente por la distribución del almacenamiento y procesamiento en nodos conectados por red y agrupados en lo que llamamos **clúster**. Hacer crecer un clúster consiste en añadir más nodos.

Los nodos con los que se construyen clústers no son necesariamente hardware especializado como sí encontramos en los superordenadores. En su lugar suelen ser equipos potentes pero que están disponibles en tiendas al alcance de cualquier usuario doméstico. A este tipo de hardware se le denomina **commodity hardware**.

La computación distribuida aporta ventajas frente a la computación centralizada en un único equipo.

Debido a que todos los nodos aportan poder de procesamiento y capacidad de almacenaje el clúster resultante tiene una mayor velocidad a la hora de realizar cálculos que si lo comparamos con una sola máquina, a esto se le conoce como **alto rendimiento**.

Disponer de un número elevado de nodos permite el **equilibrado de carga** haciendo que los trabajos se distribuyan entre nodos en función de la carga de los nodos, de la distancia física entre nodos o incluso de la disponibilidad de hardware especializado.

Usar un número grande de commodity hardware hace que el clúster sea propenso a fallos hardware. Gracias a que los datos están replicados y que los nodos están monitorizados se puede detectar la caída de algún nodo y tratar de volver a arrancarlo o delegar sus tareas en otros. El usuario debería estar informado de que el sistema está funcionando en modo degradado, pero no debería notar prácticamente nada en su funcionamiento. A esta característica se le llama **alta disponibilidad** en alusión a que los datos siempre están disponibles incluso cuando un nodo falla.

Otra de las ventajas de los clústers es la **escalabilidad** que, como hemos visto anteriormente, puede ser horizontal o vertical. Que un clúster sea escalable nos facilita hacer un cálculo más relajado de los requerimientos iniciales y también nos brinda la opción de ampliarlo o rebajarlo según las necesidades.

La replicación de datos entre nodos o los nodos de backup tienen como objetivo que el sistema se **tolerante a fallos**. Este término y alta disponibilidad son prácticamente iguales. Normalmente usamos tolerante a fallos a sistemas pequeños o nodos por ejemplo cuando hablamos de un RAID.

Los clústers también tienen una serie de desventaja que conviene tener presente.

En primer lugar, es que el suele usar **hardware heterogéneo** haciendo que su administración y mucho más la automatización sea complicada.

La **gestión de recursos** es complicada y puede haber problemas de **sincronización** difíciles de detectar.

Tal vez el mayor problema de los clústers tiene que ver con la **privacidad y la seguridad** de los datos que allí se guardan. El sistema de ficheros HDFS no cuenta con ningún tipo de cifrado y su seguridad se basa en permisos de lectura y escritura por usuarios.

3.- Conceptos de almacenamiento de datos

3.1.- Base de datos relacional

Una base de datos relacional es aquella que cumple con el modelo relacional de Codd. En estas bases de datos existe un esquema que define las relaciones entre los datos que se almacenan en tablas. A su vez define estas tablas mediante campos de tipos de datos fijos y conocidos.

Una base de datos relacional tiene sentido cuando conocemos completamente el esquema de los datos. Por ejemplo, en una biblioteca tradicional solo hay socios, libros y préstamos. Cada una de estas 3 entidades está claramente definida en una tabla con todos los campos que componen un registro. Además, en el esquema aparecerá claramente que el préstamo será lo que una las dos tablas.

3.2.- Dataset

Un dataset es un conjunto de datos relacionados. Normalmente tienen un origen común como una estación meteorológica para unos datos de temperatura o un gran almacén para ventas. En general los datasets están bien estructurados y los datos se presentan tabulados donde cada columna representa una variable concreta y cada fila representa un único evento.

Los dataset los podemos encontrar en forma de ficheros, bases de datos o incluso sitios web accesibles mediante una URL. Los dataset también los podemos generar nosotros simplemente anotando los datos como si fuera un log, usando hojas de cálculo, bases de datos, web scraping, herramientas de extracción de datos, APIs o aplicaciones de terceros.

3.3.- Almacén de datos

Los datawarehouse o almacenes de datos ya los hemos definido como repositorios de datos procesados y estructurados que usaremos para hacer analítica.

3.4.- ACID

ACID en bases de datos es el acrónimo que resume el comportamiento general esperado en una base de datos. Cumplir con estas características no es sencillo y requiere en muchos casos de escrituras en registros provisionales, bloqueos y copias de datos que lastran el rendimiento de la base de datos.

Atomicidad. Es la propiedad de las bases de datos que nos garantiza que las operaciones o se hacen completamente o dejan el sistema como antes de ser ejecutadas, es decir, nunca dejan el sistema

inconsistente por una operación hecha a medias. Nunca dejarán un registro a medias, o lo añaden entero o queda la base de datos como antes de empezar la operación. "Todo o nada".

- **Consistencia**. O también llamada integridad. Esta propiedad garantiza que la base de datos pasa de un estado válido a otro estado válido y que ninguna aplicación dejará la base de datos rota. Imagina una inserción de un registro sin clave primaria donde es necesaria, o con clave repetida, ...
- **Aislamiento**. (Isolated en inglés). Esta propiedad garantiza que el resultado de una operación no se verá afectado por ninguna otra operación que se esté dando en la base de datos. Imagínate que mientras se está eliminando un registro de una tabla, otro usuario está modificando cualquier otro registro. Si la base de datos cumple con ACID sabremos que estas dos operaciones se pueden realizar simultáneamente y que no habrá problemas. También es interesante cuando hay que aplicar varias operaciones concatenadas al mismo dato. El aislamiento permite asegurar que hasta que no acabe una operación no empezará la siguiente.
- **Durabilidad**. También llamado persistencia. Con esta propiedad garantizamos que una vez que la operación se realice los cambios serán permanentes y no se podrán deshacer. En bases de datos ACID no encontraremos un botón guardar en el que pinchar después de cada cambio.

3.5.- Teorema CAP

El teorema CAP o conjetura de Brewer hace referencia a los sistemas distribuidos y se aplica a nuestros clústers y especialmente a las aplicaciones como las bases de datos.

CAP es un acrónimo de 3 características deseables en sistemas distribuidos.

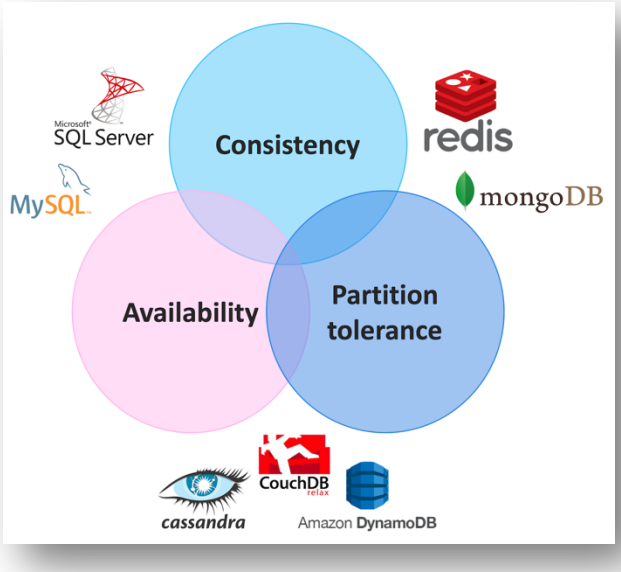
- **Consistencia**. (Consistency) Esta característica indica que todos los clientes tienen una misma imagen de los datos con independencia del nodo al que estén conectados. En principio el dato será el más reciente o bien un error.
- **Disponibilidad**. (Availability) Esta característica expresa que cualquier cliente recibirá una respuesta a su consulta con independencia de si algún nodo está caído. Con esta característica no hay errores, siempre hay una respuesta, aunque no necesariamente la más actual.
- **Tolerancia a la partición**. (Partition Tolerance) Con esta característica se indica que el sistema seguirá funcionando incluso si hay una interrupción o retraso en las comunicaciones entre nodos de manera que algunos de los mensajes no lleguen a nodo destino.

El teorema CAP dice que solo se puede asegurar simultáneamente 2 de las 3 características y para ello se establecen varias agrupaciones.

- **CA (Consistencia y Disponibilidad).** Aquí tenemos a las bases de datos relacionales en las que está garantizado que todos los clientes ven lo mismo y que siempre hay una respuesta válida.

El resto de las bases de datos son distribuidas, almacenadas en varios nodos. Es seguro que la red va a fallar en algún momento así que la característica de tolerancia a la partición es obligatoria. Tendremos que elegir con cuál de las otras dos características nos quedamos.

- **CP (Consistencia y Tolerancia a partición).** En estas bases de datos distribuidas queremos exactitud y estamos dispuestos a mandar un mensaje de error si se da el caso de que el nodo no puede garantizar estar actualizado.
- **CA (Consistencia y Disponibilidad).** Estas bases de datos distribuidas siempre dan una respuesta, aunque tal vez no precisa por estar sin actualizar. El número de seguidores o el número de likes no es un dato que requiera una exactitud tremenda y en la mayoría de los casos aceptamos un dato "aún no actualizado" como respuesta a una consulta.



3.6.- BASE

BASE es un acrónimo que resume un principio de diseño de bases de datos distribuidas teniendo en cuenta las limitaciones que impone el teorema CAP. En este caso, la tolerancia a particiones es obligatoria por ser distribuida y se prefiere la disponibilidad por encima de la consistencia, es decir, una AP según en teorema CAP.

El acrónimo viene de Basically Available, soft state & eventual consistency.

Los resultados de lecturas pueden no ser consistentes porque tal vez estemos leyendo de un nodo aun no actualizado. El estado blando hace referencia a que es posible que dos lecturas consecutivas devuelvan resultados distintos incluso sin operaciones entre lecturas, puede que en ese momento se haya actualizado el nodo. Eventualmente consistente indica que una escritura en la base de datos puede tardar un tiempo en propagarse y que durante este tiempo la base de datos se encontrará en estado blando.

4.- Conceptos de procesamiento de datos.

Entender cómo funciona un ordenador es vital para entender en detalle qué significa el procesamiento paralelo. En la arquitectura de Von Neumann existe una memoria principal donde se almacenan datos e instrucciones. Esta memoria está conectada a una CPU que será la encargada de decodificar y ejecutar las instrucciones.

En la CPU es donde tiene lugar el procesamiento de los datos y se realiza en varias fases.

1. Traer la instrucción de memoria
2. Decodificarla para saber qué circuitos activar en esa operación
3. Traer de memoria los datos que hicieran falta según la operación.
4. Ejecutar la operación
5. Guardar el resultado en memoria

En los equipos actuales, los procesadores tienen varios núcleos con varios hilos de ejecución. Esto quiere decir que tienen unidades funcionales del procesador duplicadas y por lo tanto podrían estar decodificando dos instrucciones a la vez.

4.1.- Procesamiento en paralelo

El procesamiento paralelo consiste en resolver un problema usando dos o más procesadores mientras se comparte una única memoria principal donde se dejan los resultados parciales hasta obtener el definitivo.

El procesamiento paralelo se hace dentro de una máquina ya que la comunicación entre los procesos se realiza usando memoria principal compartida.

Algunos problemas admiten una división en partes fácilmente. Calcular cuantos números primos hay en un rango o sumar muchos números son tareas que se pueden dividir en partes y finalmente unir los resultados parciales.

En Big Data hay procesamiento paralelo en los nodos ya que en ellos se ejecutan varios servicios y pueden ser asignados a varios procesadores del nodo.

4.2.- Procesamiento distribuido

El procesamiento distribuido es muy similar al procesamiento paralelo de hecho, en muchas ocasiones, por abuso del lenguaje, usamos uno u otro indiferentemente.

La diferencia fundamental es que la comunicación entre procesos ya no se realiza en una memoria principal compartida. Esto permite que los procesadores puedan estar ubicados en máquinas distintas, racks distintos o incluso en ubicaciones distintas.

En Big Data se realiza fundamentalmente un escalado horizontal aumentando el número de nodos para tener un clúster con más capacidad de proceso. Por esta razón el procesamiento distribuido es el que encontramos cuando lanzamos una tarea en el clúster hadoop, se resuelve en nodos independientes.

En el procesamiento distribuido no se usa una memoria común pero sí se siguen enviando mensajes para controlar la concurrencia. Por ello, a la hora de asignar las partes a los nodos se tiene en cuenta si la comunicación entre nodos se establece en mismo switch, en distintos racks o incluso en distintos CPD. Cuanto más lejos estén los nodos, más lenta será la comunicación.

4.3.- Estrategias de procesamiento de datos

En Big Data los datos llegan en grandes variables y en velocidades distintas. Con todos esos datos se generan unos resultados que serán servidos al usuario con mayor o menor premura.

Batch. También llamado procesamiento por lotes. Esta estrategia consiste en acumular una cantidad de datos y procesarlos en bloque. Normalmente usaremos esta estrategia cuando la velocidad de respuesta en la presentación de resultados no sea nada importante. Un ejemplo sería el renderizado de una película donde es fácil particionar el trabajo para procesarlo de manera distribuida, además no requiere demasiada comunicación entre nodos y el resultado es aceptado minutos u horas después.

Transaccional. Este tipo de procesamiento se suele dar en el escenario completamente inverso al batch. Aquí los datos que se almacenan son pequeños y en velocidad constante por ello el resultado debe obtenerse rápido. Los ejemplos típicos son todos aquellos relacionados con los registros en una base de datos. El peso de un nuevo registro es pequeño y prácticamente fijo por ello, la velocidad a la que se transmite la información es constante. Al realizar cualquier operación con una base de datos queremos que el resultado, que quede guardado el registro con los cambios, sea una operación rápida. No sería bueno recopilar un montón de cambios y fijarlos en la base de datos ya que entre esos momentos puede que haya una consulta y los datos no actualizados.

En tiempo real. Es una estrategia muy similar a la transaccional en cuanto al tiempo de respuesta mínimo pero el enfoque es sutilmente distinto. En transaccional es muy importante dar el resultado correcto porque el funcionamiento coherente del sistema depende de ello. En tiempo real la respuesta totalmente correcta, es decir, no totalmente actualizada, es menos importante. Encontramos esta estrategia en analítica de datos como por ejemplo en mostrar una gráfica de nuevos seguidores o likes a un post. Fíjate que en analítica leemos datos y generamos información, en realidad no hay transacción por lo que en este tipo de estrategia de procesamiento se suele trabaja en memoria para acelerar lo máximo posible el proceso.

Streaming. Una variante de la estrategia en tiempo real es el streaming. La mayor diferencia es que los datos llegan en forma de flujo que puede ser constante o variable. Piensa que no podremos esperar a que llegue todo el archivo, es necesario interpretarlo y procesarlo a medida que va llegando. En una retransmisión de vídeo de 2 horas habría que esperar esas dos horas para procesar la multimedia, en cambio con la estrategia de procesamiento en streaming se pueden ir analizando las partes a medida que van llegando. Esta estrategia es mucho más compleja puesto que requiere de herramientas que sepan interpretar esos flujos.

4.4.- OLTP

Acrónimo de On Line Transaction Processing. Agrupa todas las tecnologías hardware y software enfocadas a al procesamiento de las **transacciones**, entendiendo estas como entradas, salidas, modificaciones y eliminaciones de datos. En general la parte software del tipo OLTP sigue el modelo de cliente servidor.

4.5.- OLAP

Acrónimo de On Line Analytical Processing. Agrupa las tecnologías dedicadas a la **consulta** de grandes cantidades de datos para obtener información. Se centra en operaciones tipo SELECT sobre bases de datos. Es la herramienta fundamental en minería de datos.

4.6.- Principio SCV

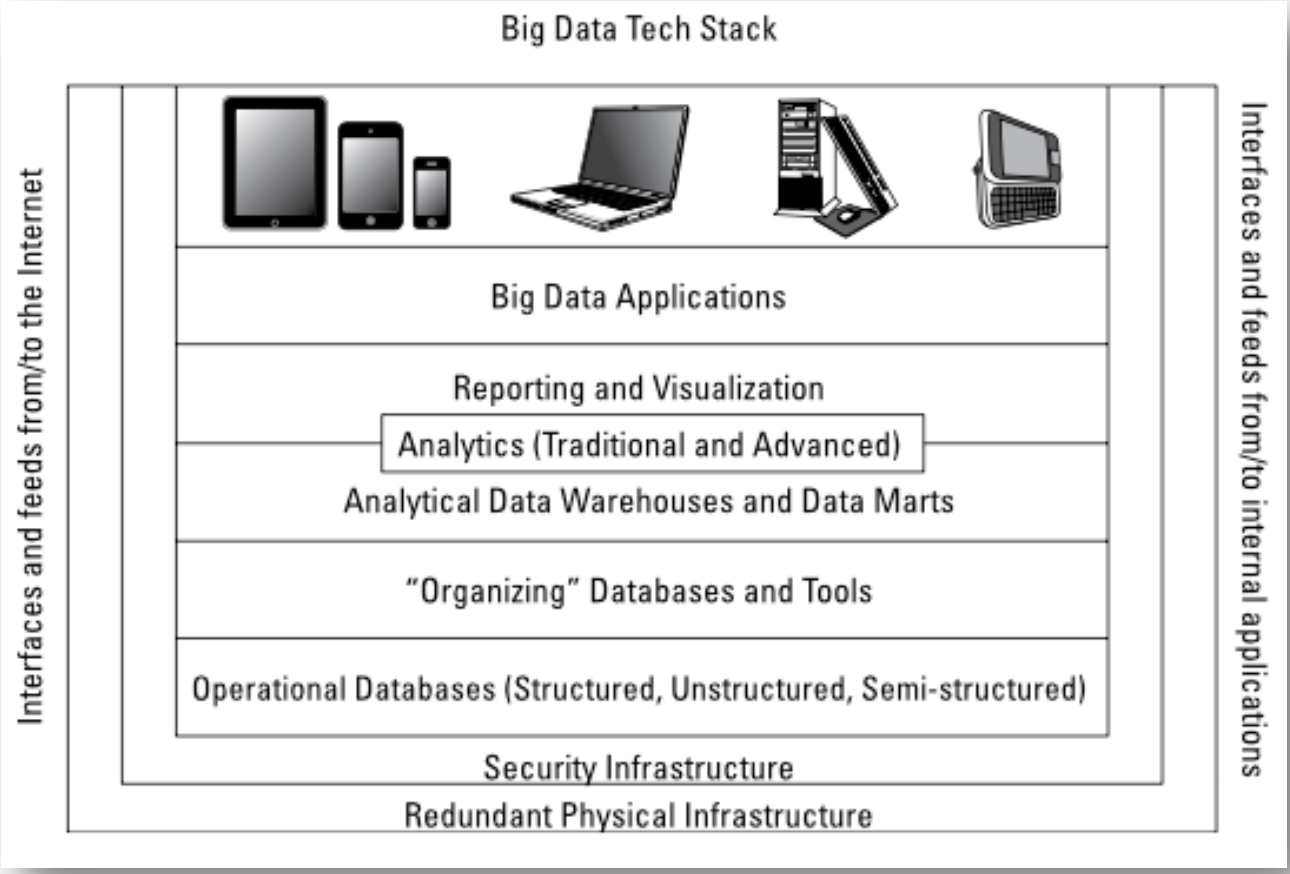
El principio SCV se aplica al procesamiento distribuido y describe 3 características de las que solo se pueden cumplir 3 simultáneamente.

- **Velocidad** (Speed). Indica el tiempo que pasa desde que los datos son recibidos por el sistema de analítica hasta que producen resultado.
- **Consistencia**. Indica la exactitud de los resultados de la analítica. Puede ser exacta utilizando todos los datos o recortar el tiempo necesario de proceso mediante técnicas de muestreo seleccionando sólo un subconjunto de datos.

- **Volumen**. Hace referencia a la cantidad de datos que pueden ser procesados.

Un sistema OLAP encargado de hacer el análisis de datos para mostrar información solo puede cumplir con dos de estas tres características que, en realidad son consecuencias unas de otras, por ejemplo, un sistema muy rápido (velocidad) y que use todos los datos para el análisis (consistencia) necesariamente no puede operar con una cantidad grande de datos (volumen)

5.- La arquitectura por capas de Big Data



En la parte de Big Data, la capa 0 es la que representa la infraestructura física redundante. Aquí se gestiona todo lo referente al almacenamiento y gestión de red buscando priorizar una serie de criterios como:

- Maximizar el **rendimiento** del hardware y de la red. En algunas ocasiones podrás verlo escrito a la inversa minimizando la latencia.

- **Disponibilidad**. Queremos que hardware y red se puedan usar en todo momento.
- **Escalabilidad**. Nos interesa que el hardware sea fácil y rápidamente sustituido o ampliado.
- **Flexible**. Queremos una infraestructura que permita añadir más nodos rápidamente si fuera necesario por lo que la estructura física de la red debe estar bien planificada. Si usamos algún tipo de RAID también será necesario contemplar el mecanismo con el que ampliaremos o sustituiremos discos.
- **Coste**. En un clúster se puede gastar todo el dinero que se tenga, pero en la mayoría de las ocasiones tendremos que decidir si queremos permitirnos el mejor equipamiento de red a costa de tener el almacenaje en discos peores que se estropeen más rápido o al revés.

Dentro de la capa 0 podemos imaginar dos ramas: la red y el almacenamiento. En cuanto a la red hay que planificar un tráfico realmente grande de datos tanto interno dentro del clúster como externo a clientes y otros clústers por ello las redes deberían tener sistemas redundantes que puedan absorber los picos de tráfico que se produzcan. En cuanto a la parte de almacenamiento también es necesario planificarlos con cuidado, una selección de discos lentos creará un cuello de botella y una mala selección de RAID también.

La capa 1 debería tratar la seguridad y privacidad. Por ser sutil diré que esta capa es la que representa el mayor desafío del Big Data. El acceso a los datos en bruto es similar al de muchos sistemas operativos por lo que será necesario crear distintos usuarios según su tipo de permiso. La encriptación es todavía un reto. En este aspecto también habría que tener en cuenta la detección de amenazas como intentos de conexión no autorizados.

La capa 2 trata de las bases de datos operacionales que son aquellas que fundamentalmente guardan registros. Estas bases de datos tienen un comportamiento descrito bajo las siglas ACID que veremos después.

En la capa 3 se encuentran los servicios de organización de datos y sus herramientas. Aquí tiene lugar lo que conoceremos como ETL que es el proceso de capturar datos, transformarlos y cargarlos en disco para que puedas ser procesados. Si alguna vez te has peleado con un PDF para importarlo en una hoja de cálculo, has estado haciendo ETL.

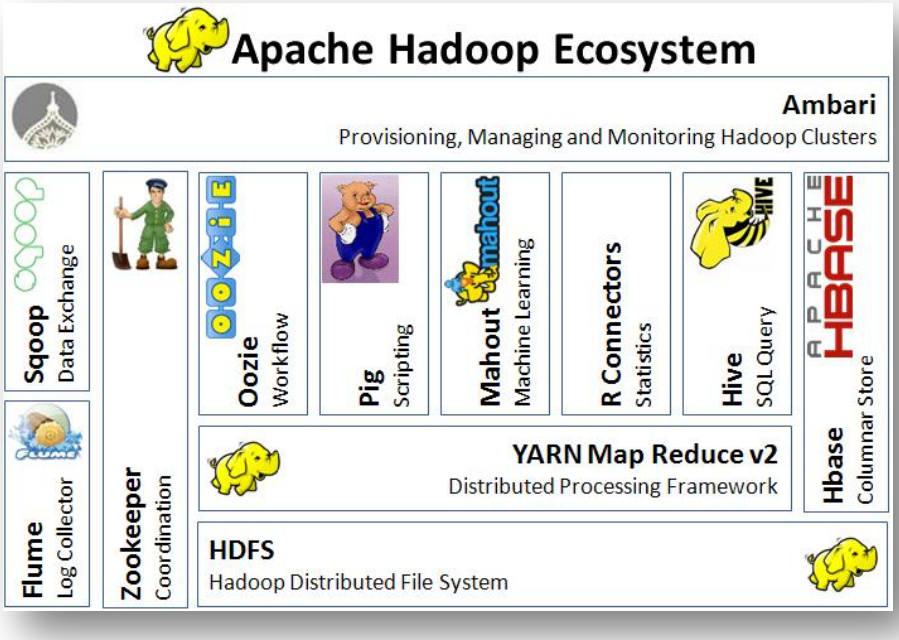
La capa 4 gestiona los almacenes de datos donde se guardan de manera distribuida todos los datos que componen el big data. Tienen distintos nombres en función del tipo de dato que guardan y el uso que se busca.

- Existen los **almacenes de datos o Data Warehouse** que son los encargados de guardar información estructurada que probablemente ha sido previamente procesada en un ETL. Es un repositorio de datos ideal para hacer analítica puesto que los datos estructurados nos recuerdan mucho a hojas de cálculo para hacer gráficas o bases de datos para hacer informes.

- En ocasiones los almacenes de datos tienen mucha más información de la necesaria. Por ejemplo, cuando pretendemos hacer analítica de ventas online igual no es necesaria toda la información de los empleados como sus fichajes o su sueldo. En estos casos se habla de **data mart**. Podríamos definirlo como un subconjunto de un data warehouse que almacena información relevante para un problema. En realidad, ambas usan la misma tecnología solo cambia la cantidad y uso que se les da a esos datos.
- También contamos con **data lakes** que son similares a los data warehouse con la diferencia que almacenan datos no estructurados y datos en bruto sin procesar. En los data lakes la filosofía es primero almacena y ya procesarás después.

Sobre estas capas tendremos dos partes fundamentales: el análisis y las aplicaciones. En el análisis tratamos de conseguir una visualización de los datos que den sentido a big data. Las aplicaciones hechas por terceros resuelven problemas de campos de la industria sanitaria, de transporte, etc.

En el módulo de Big Data Aplicado veremos y estudiaremos fundamentalmente Hadoop que tiene su propia arquitectura de capas equivalente a la anterior.



6.- El paisaje de Big Data

Llamamos paisaje de Big Data a las tendencias y conjunto de aplicaciones que hacen uso del Big Data. Incluso antes de la llegada del término Big Data ya había empresas y herramientas que estaban trabajando en este campo. Por ejemplo, el sistema de ficheros NFS es un sistema distribuido de ficheros que se creó en 1984. Otro ejemplo es la empresa Teradata, creada en 1979 se especializó en el

almacenamiento masivo distribuido desarrollando el primer sistema como más de un 1 terabyte para los almacenes Wal-mart en 1992. Piensa que en ese momento los ordenares eran 486 con un disco de unos 200 megabytes.

Principios de la década 2010 fue una explosión de nuevas herramientas y equipamiento cada vez más asequible. Los ordenadores entraron en las casas para quedarse y todo el mundo tuvo un acceso total a la tecnología. En este momento nace Hadoop como familia de herramientas que daba soporte al Big Data y la inteligencia artificial.

El movimiento Linux también se extendió entre los usuarios. La cultura colaborativa hacía que hubiera una alternativa libre a casi todas las herramientas de pago que existían en ese momento. Ejemplos hay muchos como LibreOffice a Microsoft Office o Gimp a Photoshop.

La web 2.0 acelera haciendo que los usuarios compartan no artículos de texto que ocupan poco sino también audios, fotos y vídeo. Las empresas se dan cuenta del potencial de toda esa información y la compran. Un ejemplo de esto lo podemos ver entre TomTom, la empresa que en ese momento tenía la hegemonía en dispositivo y mapas basados en posiciones GPS y la empresa de telecomunicaciones Vodafone. Tomtom compraba a Vodafone la posición de sus abonados totalmente anonimizada para detectar cuando había mucha acumulación de abonados en un tramo de carretera. De esta manera detectaba atascos y proponía rutas alternativas.

Empresas como Amazon, Microsoft y Google ya trabajan con grandes centros de datos donde disponen de una capacidad de almacenamiento y procesamiento muy importante, tanto que la empiezan a alquilar en lo que se ha denominado nube.

Con la nube disponible y sin tener que preocuparse por la parte de almacenamiento y proceso nacen muchas otras empresas que se dedican a proporcionar servicios sobre la infraestructura de los primeros. Así Instagram, Netflix, Spotify o LinkedIn están completamente alojados en Amazon AWS.

Surgen muchas empresas capaces de manejar mucha información porque disponen de herramientas de pago, pero también gratuitas para montar sus propios clústers de datos.

Existen distribuciones gratuitas para montar clústers Hadoop como Cloudera y Hortonworks que cualquiera puede descargarse y montar en su propia infraestructura privada.

Desde finales de la década de 2010 y principios de 2020 muchas de las herramientas que surgieron en la primera oleada han dejado de recibir soporte por parte de la comunidad y se mantienen funcionando, pero congeladas. Algunas de las herramientas llevan sin actualizarse desde 2016 y siguen funcionando

con Java 8 u 11 en el mejor de los casos. Las distribuciones Linux con Hadoop que antes eran gratuitas pasaron a ser de pago, como Cloudera tras comprar y absorber a Hortonworks.

En este momento parece que el mercado se está decantando a usar los sistemas que ya tienen implementados Microsoft, Google y especialmente Amazon. Allí puedes crear máquinas virtuales para montar lo que necesites. También todas estas empresas disponen de servicios ya implementados que evitan tener que crear las máquinas virtuales.

Durante este curso es poco probable que podamos ver las plataformas de Amazon, Microsoft o Google porque, aunque todas tienen una capa gratuita, es necesario registrarse con un medio de pago por si excedemos el consumo de tráfico, espacio o proceso. Lo que sí es posible que podamos ver y usar la infraestructura del Centro de Supercomputación de Galicia (CESGA) que nos permitirá hacernos una idea de su funcionamiento.

El objetivo fundamental es conocer las herramientas que sirven de base para trabajar en Big Data como Hadoop, conocer las actuales como Spark y entender el surgir de nuevas como las bases de datos NewSQL.